

Prova scritta del 11/1/2011

Strutturare adeguatamente i programmi ed evidenziarne la strutturazione mediante indentazione. Inserire anche adeguati commenti

1) (a) Realizzare una funzione di nome `compatta` che, presi come suoi parametri un array di numeri reali `A`, il numero `n` di elementi in `A` ed un array di numeri reali `B`, copia ordinatamente (dal primo all'ultimo) in `B` tutti e soli i numeri di `A` che differiscono da quelli già inseriti in `B` per almeno `0.5` (in `+` o in `-`). La funzione restituisce il numero di elementi memorizzati in `B`.

Esempio.

```
A = {1.3, 2.7, 3.12, 0.81, 0.75, 5.55} (n = 6)
B = {1.3, 2.7, 0.75, 5.55}
```

(b) Scrivere un programma principale di prova che crea ed inizializza un array `V1` di 5 elementi come nell'esempio di sopra e quindi scrive su std output il nuovo array ottenuto da `V1` tramite la `compatta`.

(c) Descrivere la funzione `compatta` anche mediante diagramma di flusso.

2) Scrivere un programma principale che legge da un file di nome `testo.txt` un testo qualsiasi e lo copia in un file di nome `testo_codificato.txt`, scambiando tra loro tutte le occorrenze di due caratteri specificati dall'utente. Controllare anche l'esistenza del file di input.

Esempio.

```
File testo.txt:           Questo e' un testo di prova.
Caratteri da scambiare:  eo
File testo_codificato.txt: Quoste o' un toste di preva.
```

Interazione con l'utente (input sottolineato):

```
Dai i due caratteri che vuoi scambiare (senza spazi): eo
Copiati 28 caratteri - fatte 6 sostituzioni
```

3) Sia `spese` una matrice le cui righe rappresentano i `giorni` di una settimana e le cui colonne (max. 10) rappresentano le `spese` effettuate durante la settimana. Ciascun elemento della matrice è costituito da una `causale di spesa` (una stringa di max. 16 caratteri) ed un `importo` (un numero reale). Scrivere un programma principale che legge da std input la causale e l'importo per ciascuna spesa effettuata, per ciascun giorno della settimana. Se la causale è "stop", l'input per il giorno corrente termina e si passa al giorno successivo. Il programma quindi calcola e stampa la somma degli importi degli elementi di ciascun giorno relativi ad una specifica causale fornita in input dall'utente. SUGG. Si definisca un tipo `struct` di nome `spesa` composto da due campi, causale di tipo stringa e importo di tipo float, e si definisca `spese` come una matrice di elementi di tipo `spesa`. N.B. Utilizzare soltanto stringhe "tipo C".

Esempio di interazione con l'utente (caso semplificato con matrice 2 x 3).

```
Dai causale e importo per ciascuna spesa del giorno 1
Causale (stop per smettere): cibo
Importo: 15.5
Causale (stop per smettere): auto
Importo: 30
Causale (stop per smettere): cibo
Importo: 21.5
```

```
Dai causale e importo per ciascuna spesa del giorno 2
Causale (stop per smettere): auto
Importo: 40
Causale (stop per smettere): stop
```

```
Dai causale di cui calcolare la somma: cibo
```

```
Somma spese per cibo nel giorno 1 : 37
Somma spese per cibo nel giorno 2 : 0
```